

## **REMARKS**

In the Official Action mailed on **28 November 2006**, the Examiner reviewed claims 1-21. The disclosure was objected to because of informalities. Claims 4, 11, and 18 were objected to as being of improper dependent form. Claims 1-2, 4-9, and 11-14 were rejected under 35 U.S.C. §102(a) as being anticipated by Leroy, (*Bytecode Verification on Java Smart Cards*, hereinafter “Leroy”). Claims 3 and 10 were rejected under 35 U.S.C. §103(a) as being unpatentable over Leroy, in view of Agesen et al (*Garbage Collection and Local Variable Type-Precision and Liveness in Java™ Virtual Machines*, hereinafter “Agesen”). Claims 15-16 and 18-21 were rejected under 35 U.S.C. §103(a) as being unpatentable over Leroy, in view of Agesen (USPN 6,047,125, hereinafter Agesen ‘125). Claim 17 was rejected under 35 U.S.C. §103(a) as being unpatentable over Leroy, in view of Agesen ‘125, and further in view of Agesen.

### **Objections to the disclosure**

The disclosure was objected to because of informalities.

Applicant has amended paragraph [0002] to update the serial number and filing data of the related application.

### **Objections to the claims**

Claims 4, 11, and 18 were objected to as being of improper dependent form.

Applicant has canceled dependent claims 4, 11, and 18 without prejudice.

### **Rejections under 35 U.S.C. §102(a) and 35 U.S.C. §103(a)**

Dependent claims 3, 10, and 17 were rejected as being unpatentable over Leroy, in view of Agesen ‘125, and further in view of Agesen. Examiner avers that Agesen teaches *wherein transforming the code module involves ensuring that*

*the evaluation stack includes only elements related to a bytecode that may trigger garbage collection when the bytecode is executed* at page 271, middle of column 2.

Applicant respectfully points out that, while Agesen teaches restricting the implementation so that collection can occur only at certain gc points (see Agesen, page 271, middle of column 2), Agesen does not teach transforming the code module to ensure that the evaluation stack includes only elements related to a bytecode that may trigger garbage collection when the bytecode is executed.

In contrast, the present invention transforms the code module, which involves ensuring that the **evaluation stack includes** only elements related to a bytecode that may **trigger garbage collection** when the bytecode is executed (see paragraph [0037] of the instant application). This is beneficial because it ensures that only parameters related to the method call are on the stack when the method is invoked, thus simplifying garbage collection and program verification.

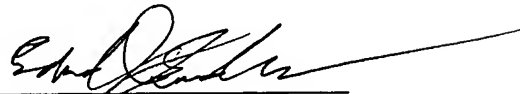
Accordingly, applicant has amended independent claims 1, 8, and 15 to include the limitations of dependent claims 3, 10, and 17, respectively, to clarify that the present invention transforms the code module, which involves ensuring that the evaluation stack includes only elements related to a bytecode that may trigger garbage collection when the bytecode is executed. These amendments find support in paragraph [0037] of the instant application. Dependent claims 3, 10, and 17 have been canceled without prejudice.

Hence, Applicant respectfully submits that independent claims 1, 8, and 15 as presently amended are in condition for allowance. Applicant also submits that claims 2 and 5-7, which depend upon claim 1, claims 9 and 12-14, which depend upon claim 8, and claims 16 and 19-21, which depend upon claim 15, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

**CONCLUSION**

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By   
Edward J. Grundler  
Registration No. 47,615

Date: 20 December 2006

Edward J. Grundler  
PARK, VAUGHAN & FLEMING LLP  
2820 Fifth Street  
Davis, CA 95618-7759  
Tel: (530) 759-1663  
Fax: (530) 759-1665  
Email: edward@parklegal.com